NASA/CR—2003-212360 MTI 95TM6

# Source Listings for Computer Code SPIRALI

## Incompressible, Turbulent Spiral Grooved Cylindrical and Face Seals

Jed A. Walowit
Jed A. Walowit, Inc., Clifton Park, New York

Wilbur Shapiro
Mechanical Technology, Inc., Latham, New York



October 2005

## The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2003-212360 MTI 95TM6

# Source Listings for Computer Code SPIRALI

## Incompressible, Turbulent Spiral Grooved Cylindrical and Face Seals

Jed A. Walowit
Jed A. Walowit, Inc., Clifton Park, New York

Wilbur Shapiro
Mechanical Technology Inc., Latham, New York



Prepared under Contract NAS3–25644

National Aeronautics and
Space Administration

Glenn Research Center

October 2005

## Document History

The source and executable files for the CFD Seal Analysis Industrial Codes, of which this is a part, were released as LEW–16582 in 1998. This report was originally published by Mechanical Technology, Inc., in March 1995.

## Document Availability Change Notice

NASA/CR—2003-212360

# Source Listings for Computer Code SPIRALI Incompressible, Turbulent Spiral Grooved Cylindrical and Face Seals

Jed A. Walowit and Wilbur Shapiro

This document was published in July 2003 with the restriction Export Administration Regulations (EAR). It was changed to Unclassified/Unlimited July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.



This work was sponsored by the Low Emissions Alternative Power Project of the Vehicle Systems Program at the NASA Glenn Research Center.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

```
C ***************************************************************************
C UPDATED INTERIM VERSION OF SPIRALI AS OF 3/9/95 THAT INCLUDES
C LOCAL PRESSURE JUMPS AT GROOVE-RIDGE INTERFACES.  ALTHOUGH THIS
C CODE DOES PROVIDE INPUT FOR CARRYOVER EFFECTS IT IS INCOMPLETE IN THAT
C IT DOES NOT FULLY IMPLEMENT THEM AS OF YET.
C ***************************************************************************

C MAIN PROGRAM FOR COMPUTER CODE SPIRALJ WHICH IS EXTENDS SPIRALI TO
C INCLUDE LOCAL PRESSURE JUMPS AT GROOVE-RIDGE DISCONTINUITIES
C FL /Gt1024 SPIRALI.FOR
      PARAMETER (NDZ=201,NDREG=21)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER TITLE*64,PNAME*60,FNAME(3)*60
      DIMENSION NRSUB(NDREG),ELFR(NDREG),RH2O(2)
      DIMENSION ALPI(NDREG),BETI(NDREG),DELT(NDREG),DELTI(NDREG)
      DIMENSION NSG(NDREG),ENGP(NDREG),ZETG(NDREG)
      DIMENSION RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION TAU(NDZ,NDREG),CK(4,4),CB(4,4),AM(4,4)
C THIS COMMON BLOCK PASSES DATA TO USER DEFINABLE FUNCTION FLMSHP
      COMMON/BFSHP/HTAP,HBRL
C THIS COMMON BLOCK PASSES TURBULENCE COEFFICIENTS TO
C USER DEFINABLE FUNCTIONS FA AND FB
      COMMON/BFAFB/EMA,ENA,EMB,ENB
      DATA RH2O/9.35726D-5,1.D3/
C INITIALIZE NAMELIST VARIABLES
      DATA TITLE,IFACE,ISIUN,IGROT,NOI,IFLOW/' ',5*0/,
     +RO,C,EL,RPM,RPM0,RPMD/6*0.D0/,
     +PLEG,PRIG,VISC,DENS/4*0.D0/,
     +FZD,IHOME,NITH,TOLH/0.D0,0,10,1.D-4/
     +NITV,TOLV,DUT/30,1.D-5,1.D-6/,
     +NREG,NRSUB(1),ELFR(1)/1,20,1.D0/,
     +ALPI,BETI,DELT,ZET/NDREG*0.D0,NDREG*0.D0,NDREG*0.D0,NDREG*0.D0/,
     +NSG,ZETG/NDREG*0,NDREG*0.D0/
      NAMELIST/INPUTS/TITLE,IFACE,ISIUN,IGROT,NOI,IFLOW,
     +RO,C,EL,RPM,RPM0,RPMD,PLEG,PRIG,VISC,DENS,EMA,ENA,EMB,ENB,
     +HTAP,HBRL,FZD,IHOME,NITH,TOLH,
     +NITV,TOLV,DUT,
     +NREG,NRSUB,ELFR,ZET,ALPI,BETI,DELT,NSG,ZETG
C USE DATA STAEMENT BELOW TO HARD CODE DEFAULT FILENAMES.  BLANK VALUES USED
C HERE CAUSES MICROSOFT COMPILER TO TAKE NAMES FROM COMMAND LINE OR ISSUE
C PROMPTS FOR THEIR INPUT AT RUN TIME.
      DATA FNAME/3*' '/
C UNIT 1 IS INPUT FILE, 2 OUTPUT FILE, 3 PLOT FILE
      OPEN(1,FILE=FNAME(1),STATUS='OLD',ERR=9999)
      OPEN(2,FILE=FNAME(2),ERR=9999)
      OPEN(3,FILE=FNAME(3),ERR=9999)
      INQUIRE(3,NAME=PNAME)
      PI=4.D0*ATAN(1.D0)
      ICASE=0
C INITIALIZE DATA ON COMMON BLOCKS
      HTAP=0.D0
      HBRL=0.D0
      EMA=-0.25D0
      EMB=EMA
      ENA=.0791D0
      ENB=ENA
    1 READ(1,INPUTS,END=999)
      ENGP=NSG/4.D0/PI
C CLEAN UP FLAGS
      IF(IGROT.NE.1)IGROT=0
      IF(IFACE.NE.1)IFACE=0
      IF(ISIUN.NE.1)ISIUN=0
C SET CASE NUMBER AND WRITE CASE NUMBER AND TITLE TO OUTPUT FILE
      ICASE=ICASE+1
      IF(ICASE.GT.1)WRITE(2,*)' '
```



```
      WRITE(2,19)ICASE,TITLE
 19   FORMAT(' ( CASE', I3,' )  ',A64/)
C PRINT OUT NAMELIST
      CALL INLIST(2,TITLE,IFACE,ISIUN,IGROT,NOI,IFLOW,
     +RO,C,EL,RPM,RPM0,RPMD,PLEG,PRIG,VISC,DENS,
     +EMA,ENA,EMB,ENB,HTAP,HBRL,
     +FZD,IHOME,NITH,TOLH,
     +NITV,TOLV,DUT,
     +NREG,NRSUB,ELFR,ZET,ALPI,BETI,DELT,NSG,ZETG)
      CALL OUTSCR('STARTING SOLUTION FOR CASE NUMBER',ICASE)
C CHECK UP FRONT FOR ERRORS
      IER=0
      IF(RO.LE.0.D0.OR.EL.LE.0.D0.OR.C.LE.0.D0.OR.VISC.LE.0.D0
     +.OR.RPM.LT.-1.D-8)IER=8
      IF(NREG.GT.NDREG)IER=10
      ELSUM=0.D0
      DO 81 K=1,NREG
      ELSUM=ELSUM+ELFR(K)
      IF(NRSUB(K).GE.NDZ)IER=9
 81   CONTINUE
      IF(ABS(ELSUM-1.D0).GT.1.01)IER=11
      IF(IER.NE.0)GO TO 80
C GENERATE R AND Z GRIDS
      ELT=EL/2.D0/RO
      CALL RZGRID(IFACE,ELT,NREG,NRSUB,ELFR,RG,ZTG)
C CHECK ON HOMING IN ON AXIAL LOAD FOR FACE SEAL
      ITH=0
      IF(IHOME.NE.2)CNEW=C
      IF(IFACE.EQ.1.AND.(IHOME.EQ.1.OR.IHOME.EQ.2)
     +.AND.FZD.GT.1.D-6)ITH=1
C LABEL BELOW IS TOP OF HOMING LOOP, USED WHEN ITH>0
 88   C1=CNEW
      IF(ITH.GT.0)CALL OUTSCR('  LOAD ITERATION NO.',ITH)
      IER=0
C GET REFERENCE GAGE PRESSURE AND CHECK DIRECTION OF POISEUILLE FLOW
      PO=PLEG-PRIG
      IDIR=1
      IF(PO.LT.0.)IDIR=-1
      PO=ABS(PO)
C CALCULATE DIMENSIONLESS PRIMARY FILM THICKNESS H, AND TOTAL NO. PTS, MTOT
      MTOT=0
      DO 7 K=1,NREG
        NRT=NRSUB(K)+1
        MTOT=MTOT+NRT
        DELTI(K)=DELT(K)/C1
        ENGP(K)=NSG(K)/4.D0/PI
        DO 7 J=1,NRT
C  ADD SHAPE AND DIVIDE BY C
        X=ZTG(J,K)/(2.D0*ELT)
        IF(IFACE.EQ.1)X=X+(ELT-1.D0)/(2.D0*ELT)
  7     H(J,K)=1.D0+DELTI(K)*ALPI(K)+FLMSHP(X)/C1
C USE SMALL NUMBER IN PLACE OF 0 DENSITY
      DNS=MAX(1.D-8*RH2O(ISIUN+1),DENS)
C CONVERT ANGULAR VELOCITIES TO RAD/SEC
      DRC=PI/30.D0
      OM=RPM*DRC
      OM0=RPM0*DRC
      OMD=RPMD*DRC
C CALCULATE VELOCITY AND REYNOLDS NUMBER FOR LAMINAR FLOW
      VPL=C1*C1/(12.D0*VISC*EL)
      VL=VPL*PO
      REL=2.D0*C1*VL*DNS/VISC
      REA=REL
      RFBR=24.D0
C GET CHARACTERISTIC AXIAL REYNOLDS NUMBER FOR TURBULENT FLOW
      IF(REA.GT.1000.D0)THEN
```

```
        CALL RECAL(REL,REA,FBAR,NITV,DUT,IER)
        IF(IER.NE.0)GO TO 99
        RFBR=REA*FBAR
      ENDIF
      VPT=VPL*24.D0/RFBR
C COMPARE AXIAL AND CIRCUMFERENTIAL REYNOLDS NUMBERS TO SELECT RE AND P0
      REO=C1*R0*OM*DNS/VISC
      IF(REA.GE.REO)THEN
         RE=REA
         PIN=1.D0
         ELSE
         TMP=P0
         P0=R0*OM/2./VPT
         PIN=TMP/P0
         RE=REO
      ENDIF
C MAKE SOME DECISIONS ON HANDLING AXIAL INERTIA
      NOI1=0
      IF(NOI.GE.0.AND.(C1/EL*REA/RFBR.LT..01.OR.NOI.GT.0))NOI1=1
      IF(IFLOW*IDIR.EQ.-1.AND.NOI1.EQ.0)THEN
        IDIR=-IDIR
        PIN=-PIN
      ENDIF
      REC=RE*2.D0*C1/R0*DENS/DNS
      IF(NOI.EQ.2)REC=0.D0
C COMPUTE CHARACTERISTIC VELOCITY AND DIMENSIONLESS PARAMETERS
      V0=RE*VISC/(2.D0*DNS*C1)
      P1R=VISC*V0*R0/(4.D0*C1*C1*P0)
      OMT=OM*R0/V0
      OM0T=OM0*R0/V0
      OMDT=OMD*R0/V0
C CALCULATE DIMENSIONLESS LOADING
      FZND=FZD/P0/R0**2
C PERFORM SEAL COMPUTATIONS
      CALL TSEAL(TOLV,NITV,NOI1,IFACE,IDIR,IGROT,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OM0T,OMDT,DUT,ZET,ALPI,BETI,DELTI,ENGP,ZETG,
     +RG,ZTG,H,U,V,P,TAU,CK,CB,AM,FLO,TOR,W,IAMASS,ITER,IER)
      PADD=0.D0
      IF(PIN.LT.0.D0)PADD=-PIN
      JEND=NRSUB(NREG)+1
      W=W+PI*PADD*(RG(JEND,NREG)**2-RG(1,1)**2)
      RE1=RE/DNS*DENS
      REA1=RE1*H(1,1)*ABS(V(1,1))
      REA2=RE1*H(JEND,NREG)*ABS(V(JEND,NREG))
      REO1=RE1*H(1,1)*ABS(U(1,1)-RG(1,1)*OMT)
      REO2=RE1*H(JEND,NREG)*ABS(U(JEND,NREG)-RG(JEND,NREG)*OMT)
C CHECK TO SEE IF RIGHT INLET BOUNDARY WAS USED
      IF(IER.EQ.0.AND.NOI1.EQ.0.AND.IFLOW*FLO.LT.0.)IER=7
C IF HOMING ON LOAD FOR FACE SEAL, CHECK FOR CONVERGENCE OR TROUBLE
      IF(ITH.GT.0.AND.IER.EQ.0.)THEN
        IF(ABS(W-FZND)/FZND.LT.TOLH)THEN
          CONTINUE
          ELSE IF(ITH.EQ.NITH)THEN
          IER=5
          ELSE IF(CK(1,1).LT.1.D-20)THEN
          IER=6
          ELSE
          CNEW=(1.D0-(FZND-W)/CK(1,1))*C1
          IF(CNEW/C1.LT.1.D-8)THEN
            IER=6
            ELSE
            ITH=ITH+1
            GO TO 88
          ENDIF
        ENDIF
      ENDIF
```



```
C WRITE OUTPUT
      NOI2=NOI1
      IF(NOI1.EQ.1.AND.REC.LT.1.D-4)NOI2=2
 80   CALL DIMOUT(2,NOI2,IFACE,ISIUN,IER,ITER,
     +W,FLO,TOR,CK,CB,AM,
     +R0,EL,C1,VISC,DENS,PLEG,PRIG,V0,P0,IAMASS,RPM,RPM0,RPMD,
     +REA1,REA2,REO1,REO2)
      IF(IER.NE.0)GO TO 99
C IF NON-NUL PLOT FILE DUMP NO. POINTS, FILM, VELOCITY AND PRESSURE DATA
      IF(PNAME.EQ.'NUL'.OR.PNAME.EQ.'nul')GO TO 99
      WRITE(3,*)MTOT
      ROM2=OM*R0/2.D0
      DO 20 K=1,NREG
       NRT=NRSUB(K)+1
 20   WRITE(3,21)(ZTG(I,K),H(I,K)*C1-ALPI(K)*DELT(K),
     +U(I,K)*V0,V(I,K)*V0,(P(I,K)+PADD)*P0,I=1,NRT)
 21   FORMAT(OP,F10.4,1P,4E13.5)
C     +U(I,K)*V0,V(I,K)*V0,(P(I,K)+PADD)*P0,RG(I,K)*ROM2,I=1,NRT)
C 21  FORMAT(OP,F10.4,1P,5E13.5)
 99   IF(IER.NE.0)CALL EMSG(IER)
      WRITE(2,*)' '
      GO TO 1
 999  CLOSE(1)
      CLOSE(2)
      CLOSE(3)
9999  STOP
      END
```

```
      SUBROUTINE EMSG(IER)
      CHARACTER*78 MSG(11)
C SENDS ERROR MESSAGES TO STD. OUTPUT.
C CALLED BY MAIN PROGRAM
      DATA MSG/
     +'INITIAL VELOCITY COMPUTATION DIVERGED',
     +'PRIMARY FLOW COMPUTATION DIVERGED',
     +'MATRIX INVERSION ERROR ENCOUNTERED IN SECOND ORDER SOLUTION',
     +'SPIRAL GROOVE LOCAL FLOW COMPUTATION DIVERGED',
     +'FACE SEAL AXIAL LOAD ITERATION DIVERGED',
     +'NEGATIVE STIFFNESS OR FILM THICKNESS IN AXIAL LOAD ITERATION',
     +'WRONG INLET BOUNDARY WAS USED WITH TRANVERSE INERTIA INLCUDED',
     +'ILLEGAL LENGTH, CLEARANCE, VISCOSITY, PRESSURE OR SPEED ENCOUNTER
     +ED',
     +'MAXIMUM NUMBER OF ALLOWABLE GRID POINTS EXCEEDED',
     +'MAXIMUM NUMBER OF ALLOWABLE REGIONS EXCEEDED',
     +'SUM OF LENGTH FRACTIONS ARE NOT EQUAL TO 1'
     +/
      WRITE(*,*)MSG(IER)
      RETURN
      END
```



```
      SUBROUTINE OUTSCR(MSG,NUM)
C SENDS STATUS MESSAGES TO THE STANDARD OUTPUT UNIT
C CALLED BY MAIN, TSEAL
      CHARACTER*(*)MSG,CNUM*6,MSG1*78
      WRITE(CNUM,'(I6)')NUM
      DO 5 I1=1,6
      I=I1
      IF(CNUM(I:I).GT.' ') GO TO 6
   5  CONTINUE
   6  CONTINUE
      L=LEN(MSG)
      MSG1=MSG
C CONCATINATE NON 0 NUMBER TO STRING
      IF(NUM.EQ.0)THEN
        LT=L
        ELSE
        LT=L+8-I
        MSG1(L+1:LT)=' '//CNUM(I:6)
      ENDIF
      WRITE(*,*)MSG1(1:LT)
      RETURN
      END
```

```
      SUBROUTINE DIMOUT(IFILE,NOI,IFACE,ISIUN,IER,ITER,
     +W,QIN,TOR,CK,CB,AM,
     +DDR,DDL,DDC,DDMU,DENS,DDPL,DDPR,V0,P0,IAMASS,RPM,RPM0,RPMD,
     +REA1,REA2,REO1,REO2)
C SENDS OUTPUT TO UNIT IFILE
C CALLED BY MAIN
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER STRK(4,2)*7,STRK0(4,2)*7,STRB(4,2)*7,
     +STRA(4,2)*7,KUNIT(4,2,2)*10,BUNIT(4,2,2)*10,AUNIT(4,2,2)*10
      CHARACTER IN(2)*6,LB(2)*6,I3S(2)*13,PSI(2)*7,IHP(2)*8,
     +XIN(2)*12,YIN(2)*12,ZIN(2)*12,ILBC(2)*10,PSI4(2)*14,PSIS(2)*12
      CHARACTER NOISTR(3)*30
      DIMENSION CK(4,4),CB(4,4),AM(4,4),SCON(4),XCON(4)
      DATA NOISTR/' ',', TRANSVERSE INERTIA NEGLECTED',
     +', INERTIA NEGLECTED'/
      DATA STRK/'Kx','Ky','Kphi','Kpsi','Kz','Kphi','Kpsi',' '/
      DATA STRK0/'K0x','K0y','K0phi','K0psi','K0z','K0phi','K0psi',' '/
      DATA STRB/'Bx','By','Bphi','Bpsi','Bz','Bphi','Bpsi',' '/
      DATA STRA/'Ax','Ay','Aphi','Apsi','Az','Aphi','Apsi',' '/
      DATA KUNIT/'LB','LB','IN-LB','IN-LB','LB','IN-LB','IN-LB',' ',
     +'N','N','N-m','N-m','N','N-m','N-m',' '/
      DATA BUNIT/'LB-SEC','LB-SEC','IN-LB-SEC','IN-LB-SEC','LB-SEC',
     +'IN-LB-SEC','IN-LB-SEC',' ',
     +'N-SEC','N-SEC','N-m-SEC','N-m-SEC','N-SEC',
     +'N-m-SEC','N-m-SEC',' '/
      DATA AUNIT/'LB-SEC2','LB-SEC2','IN-LB-SEC2','IN-LB-SEC2',
     +'LB-SEC2','IN-LB-SEC2','IN-LB-SEC2',' ',
     +'N-SEC2','N-SEC2','N-m-SEC2','N-m-SEC2','N-SEC2',
     +'N-m-SEC2','N-m-SEC2',' '/
      DATA XIN,YIN,ZIN/'   x (IN)','   x (m)','   y (IN)','   y (m)',
     +'   z (IN)','   z (m)'/
      DATA IN,LB/'  (IN)','  (m)','  (LB)','  (N)'/
      DATA ILBC/'  (IN-LB)','  (N-m),'/
      DATA I3S,PSI/'  (IN**3/SEC)','  (m**3/SEC)','  (PSI)','  (Pa)'/
      DATA PSIS/'  (PSI-SEC)','  (Pa-SEC),'/
      DATA PSI4/'  (LB-SEC/IN4)','  (Kg/m3)'/
      DATA IHP/'  (HP)','  (WATT)'/
      K=1
      IF(ISIUN.EQ.1)K=2
      IF(IFACE.EQ.1)GO TO 1000
      IF(IER.EQ.0)THEN
        FCON=P0*DDR*DDR
        QCON=V0*DDC*DDR
        TCON=DDR/V0
        HP=TOR*FCON*DDC*RPM*1.586662957D-5
        IF(K.EQ.2)HP=HP*6600
        SCON(1)=FCON
        SCON(2)=FCON
        SCON(3)=FCON*DDR
        SCON(4)=FCON*DDR
        XCON(1)=DDC
        XCON(2)=DDC
        XCON(3)=DDC/DDR
        XCON(4)=DDC/DDR
      ENDIF
      NFP=4
      WRITE(IFILE,60)NOISTR(NOI+1)
   60 FORMAT (/' CYLINDRICAL SEAL',A30/)
      WRITE(IFILE,61)DDL,2.D0*DDR,DDC,IN(K)
   61 FORMAT(' LENGTH, DIAMETER, CLEARANCE =',1P,E12.4,',',E12.4,',',
     +E12.4,A6/)
      WRITE(IFILE,62)RPM,RPM0,RPMD
   62 FORMAT(' ROTOR, SWIRL AND DIST. SPEEDS =',1P,E12.4,
     +',',E12.4,',',E12.4,'  (RPM)'/)
      WRITE(IFILE,63)DDPL,DDPR,PSI(K)
   63 FORMAT(' PRESSURE AT START, END AXIAL BOUNDARIES =',1P,E12.4,',',
```



```
     +E12.4,A7/)
      WRITE(IFILE,64)DDMU,PSIS(K),DENS,PSI4(K)
   64 FORMAT(' VISCOSITY =',1P,E12.4,A12,
     +'   DENSITY =',E12.4,A14/)
      WRITE(IFILE,40)IER,ITER
      IF(IER.NE.0)RETURN
   40 FORMAT(' ERROR CODE =',I3,',   ITERATIONS IN PRIMARY FLOW =',I3)
      WRITE(IFILE,48)QIN*QCON,I3S(K),TOR*FCON*DDC,ILBC(K),
     +HP,IHP(K),REA1,REO1,REO2
   48 FORMAT(/' FLOW =',1P,E12.4,A13//
     +' TORQUE =',1P,E12.4,A10,'   FILM POWER LOSS =',E12.4,A8//
     +' AXIAL REYNOLDS NUMBER                          =',1P,E12.4,/
     +' CIRC. REYNOLDS NUMBERS FOR ROTOR AT SEAL ENDS =',1P,E12.4,',',
     +E12.4)
      WRITE(IFILE,50)
   50 FORMAT(/'         DYNAMIC COEFFICIENTS  ( FORCE UNIT / DISP. UNIT
     +)')
      ASSIGN 45 TO KF
      WRITE(IFILE,KF)XIN(K),YIN(K)
   45 FORMAT(/' DISP.   ',2A12,
     +' phi (RAD)  ',' psi (RAD)  ','  FORCE UNIT ')
      ASSIGN 47 TO KF
      DO 100 I=1,NFP
        IF(IAMASS.EQ.1)THEN
          WRITE(IFILE,KF)STRK(I,IFACE+1),
     +    (CK(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
          ELSE
          WRITE(IFILE,KF)STRK(I,IFACE+1),
     +    (AM(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
        ENDIF
  100 CONTINUE
      DO 101 I=1,NFP
  101 WRITE(IFILE,KF)STRB(I,IFACE+1),(CB(I,J)*TCON*SCON(I)/XCON(J),
     +J=1,NFP),BUNIT(I,IFACE+1,K)
      DO 102 I=1,NFP
        IF(IAMASS.EQ.1)THEN
          WRITE(IFILE,KF)STRA(I,IFACE+1),
     +    (AM(I,J)*SCON(I)*TCON**2/XCON(J),J=1,NFP),AUNIT(I,IFACE+1,K)
          ELSE
          WRITE(IFILE,KF)STRK0(I,IFACE+1),
     +    (CK(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
        ENDIF
  102 CONTINUE
   47 FORMAT(1X,A7,1P,4E12.4,3X,A10)
      RETURN
 1000 IF(IER.EQ.0)THEN
        FCON=P0*DDR*DDR
        QCON=V0*DDC*DDR
        TCON=DDR/V0
        HP=TOR*FCON*DDC*RPM*1.586662957D-5
        IF(K.EQ.2)HP=HP*6600
        SCON(1)=FCON
        SCON(2)=FCON*DDR
        SCON(3)=FCON*DDR
        XCON(1)=DDC
        XCON(2)=DDC/DDR
        XCON(3)=DDC/DDR
      ENDIF
      NFP=3
      WRITE(IFILE,860)NOISTR(NOI+1)
  860 FORMAT (/' FACE SEAL',A30/)
      WRITE(IFILE,861)2.D0*(DDR-DDL),2.D0*DDR,DDC,IN(K)
  861 FORMAT(' ID, OD, NOMINAL FILM THICKNESS =',
     +1P,E12.4,',',E12.4,',',E12.4,A6/)
      WRITE(IFILE,62)RPM,RPM0,RPMD
      WRITE(IFILE,863)DDPL,DDPR,PSI(K)
```

```
863  FORMAT(' INSIDE, OUTSIDE PRESSURE =',1P,E12.4,',',
     +E12.4,A7/)
      WRITE(IFILE,64)DDMU,PSIS(K),DENS,PSI4(K)
      WRITE(IFILE,40)IER,ITER
      IF(IER.NE.0)RETURN
      WRITE(IFILE,144)W*FCON,LB(K)
144   FORMAT(/' AXIAL LOAD TO BALANCE FACE SEAL =',1P,E12.4,A6)
      WRITE(IFILE,848)QIN*QCON,I3S(K),TOR*FCON*DDC,ILBC(K),
     +HP,IHP(K),REA1,REA2,REO1,REO2
848   FORMAT(/' FLOW =',1P,E12.4,A13//
     +' TORQUE =',1P,E12.4,A10,'   FILM POWER LOSS =',E12.4,A8//
     +' RADIAL REYNOLDS NUMBER AT ID, OD          =',1P,E12.4,',',E12.4
     +/' CIRC. REYNOLDS NUMBERS FOR ROTOR AT ID, OD =',1P,E12.4,',',
     +E12.4)
      WRITE(IFILE,50)
      ASSIGN 145 TO KF
      WRITE(IFILE,KF)ZIN(K)
145   FORMAT(/' DISP.   ',A12,
     +' phi (RAD)  ',' psi (RAD)  ',' FORCE UNIT ')
      ASSIGN 147 TO KF
      DO 500 I=1,NFP
        IF(IAMASS.EQ.1)THEN
          WRITE(IFILE,KF)STRK(I,IFACE+1),
     +    (CK(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
          ELSE
          WRITE(IFILE,KF)STRK(I,IFACE+1),
     +    (AM(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
        ENDIF
500   CONTINUE
      DO 501 I=1,NFP
501   WRITE(IFILE,KF)STRB(I,IFACE+1),(CB(I,J)*TCON*SCON(I)/XCON(J),
     +J=1,NFP),BUNIT(I,IFACE+1,K)
      DO 502 I=1,NFP
        IF(IAMASS.EQ.1)THEN
          WRITE(IFILE,KF)STRA(I,IFACE+1),
     +    (AM(I,J)*SCON(I)*TCON**2/XCON(J),J=1,NFP),AUNIT(I,IFACE+1,K)
          ELSE
          WRITE(IFILE,KF)STRKO(I,IFACE+1),
     +    (CK(I,J)*SCON(I)/XCON(J),J=1,NFP),KUNIT(I,IFACE+1,K)
        ENDIF
502   CONTINUE
147   FORMAT(1X,A7,1P,3E12.4,3X,A10)
      RETURN
      END
```



```
      SUBROUTINE INLIST(IFILE,TITLE,IFACE,ISIUN,IGROT,NOI,IFLOW,
     +RO,C,EL,RPM,RPMO,RPMD,PLEG,PRIG,VISC,DENS,
     +EMA,ENA,EMB,ENB,HTAP,HBRL,
     +FZD,IHOME,NITH,TOLH,
     +NITV,TOLV,DUT,
     +NREG,NRSUB,ELFR,ZET,ALPI,BETI,DELT,NSG,ZETG)
C THIS ROUTINE PRINTS OUT THE NAMELIST IN A LEGIBLE MANNER
C CALLED BY MAIN PROGRAM
C WRITES TO UNIT NO. IFILE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      CHARACTER FORM*80,TITLE*64,NTC*2
      DIMENSION NRSUB(NDREG),ELFR(NDREG),ZET(NDREG)
      DIMENSION ALPI(NDREG),BETI(NDREG),DELT(NDREG)
      DIMENSION NSG(NDREG),ZETG(NDREG)
      NT=LEN(TITLE)
      DO 80 I=1,NT
      IT=NT+1-I
      IF(TITLE(IT:IT).GT.' ')GO TO 81
 80   CONTINUE
 81   WRITE(NTC,'(I2)')IT
      WRITE(IFILE,*)'&INPUTS'
      FORM='(4X,A8,2X,A1,A'//NTC//',A1)'
      WRITE(IFILE,FORM)'TITLE  =',"'",TITLE,"'"
      WRITE(IFILE,1)'IFACE  =',IFACE,'ISIUN  =',ISIUN
      WRITE(IFILE,1)'IGROT  =',IGROT,'NOI    =',NOI,'IFLOW  =',IFLOW
      WRITE(IFILE,2)'RO     =',RO,'EL     =',EL,'C      =',C
      WRITE(IFILE,2)'RPM    =',RPM,'RPMO   =',RPMO,'RPMD   =',RPMD
      WRITE(IFILE,2)'PLEG   =',PLEG,'PRIG   =',PRIG,'FZD    =',FZD
      WRITE(IFILE,2)'VISC   =',VISC,'DENS   =',DENS
      WRITE(IFILE,2)'EMA    =',EMA,'ENA    =',ENA
      WRITE(IFILE,2)'EMB    =',EMB,'ENB    =',ENB
      WRITE(IFILE,2)'HTAP   =',HTAP,'HBRL   =',HBRL
      WRITE(IFILE,2)'TOLH   =',TOLH,'TOLV   =',TOLV,'DUT    =',DUT
      WRITE(IFILE,3)'IHOME  =',IHOME,'NITH   =',NITH,'NITV   =',NITV
      NREG5=MIN(NREG,5)
      WRITE(IFILE,4)'NREG   =',NREG,'NRSUB  =',(NRSUB(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,6)(NRSUB(I),I=6,NREG)
      WRITE(IFILE,5)'ELFR   =',(ELFR(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(ELFR(I),I=6,NREG)
      WRITE(IFILE,5)'ZET    =',(ZET(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(ZET(I),I=6,NREG)
      WRITE(IFILE,5)'ALPI   =',(ALPI(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(ALPI(I),I=6,NREG)
      WRITE(IFILE,5)'BETI   =',(BETI(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(BETI(I),I=6,NREG)
      WRITE(IFILE,5)'DELT   =',(DELT(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(DELT(I),I=6,NREG)
      WRITE(IFILE,8)'NSG    =',(NSG(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,9)(NSG(I),I=6,NREG)
      WRITE(IFILE,5)'ZETG   =',(ZETG(I),I=1,NREG5)
      IF(NREG.GT.5)WRITE(IFILE,7)(ZETG(I),I=6,NREG)
 1    FORMAT(4X,A8,I3,T30,A8,I3,T55,A8,I3)
 2    FORMAT(4X,A8,1P,E12.4,T30,A8,E12.4,T55,A8,E12.4)
 3    FORMAT(4X,A8,I3,T30,A8,I4,T55,A8,I4)
 4    FORMAT(4X,A8,I3,T30,A8,5I4)
 5    FORMAT(4X,A8,1P,5E12.4)
 6    FORMAT(37X,5I4)
 7    FORMAT(12X,1P,5E12.4)
 8    FORMAT(4X,A8,5I4)
 9    FORMAT(12X,5I4)
      WRITE(IFILE,*)'/'
      RETURN
      END
```

```
      SUBROUTINE TSEAL(TOLV,NITV,NOI,IFACE,IDIR,IGROT1,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,OMDT,DUT,ZET,ALP,BET,DELT,ENGP,ZETG,
     +RG,ZTG,H,U,V,P,TAU,CK,CB,AM,FLO,TOR,W,IAMASS,ITER,IER)
C TURBULENT SEAL COMPUTATION SUBROUTINE
C CALLED BY MAIN
C CALLS VISOLV,TORQ,FORCE,KBCAL
C FLAG DEFINITIONS:
C   NOI    =   1 NEGLECT AXIAL CONVECTIVE INERTIAL TERMS
C   IFACE  =   1 FACE SEAL
C              0 CYLINDRICAL SEAL
C   IDIR   =   1 AXIAL FLOW IS KNOWN POSITIVE
C             -1 AXIAL FLOW IS KNOWN NEGATIVE
C   IGROT  =   1 GROOVES ROTATE
C              0 GROOVES STATIONARY
C             -1 NO GROOVES (SET BY THIS SUB AND PASSED TO SUPPORTING SUBS)
C   IAMASS=    1 CK, CB, AND AM ARE STIFFNESS, DAMPING AND MASS AT 0 FREQUENCY
C              0 CK AND AM ARE DAMPING AND STIFFNESS AT DISTURBANCE FREQUENCY
C IAMASS AND IER (ERROR CODE ) ARE OUTPUT
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
      DIMENSION ALP(NDREG),BET(NDREG),SBET(NDREG),CBET(NDREG),
     +DELT(NDREG),IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION TAU(NDZ,NDREG),CK(4,4),CB(4,4),AM(4,4),
     +TMP(4,4)
      IER=0
      PI=4.D0*ATAN(1.D0)
C ADJUST FLAG AND GET SIN AND COS FOR SPIRAL GROOVE REGION
      DO 5 K=1,NREG
        IF(ALP(K).LT.1.D-8.OR.ABS(1.D0-ALP(K)).LT.1.D-8.OR.
     +  ABS(BET(K)).LT.1.D-8)THEN
C IGROT=-1 SIGNIFIES NO GROOVES
          IGROT(K)=-1
          ELSE
          IGROT(K)=IGROT1
          SBET(K)=SIN(BET(K)*PI/180.D0)
          CBET(K)=COS(BET(K)*PI/180.D0)
        ENDIF
  5   CONTINUE
C HOME IN ON INITIAL VELOCITY AND GET VELOCITY AND PRESSURE DISTRIBUTIONS
      VI=ABS(PIN)
      VI=MAX(VI,1.D-3)
      CALL OUTSCR('    FIRST ORDER SOLUTION',0)
      CALL VISOLV(TOLV,NITV,VI,NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,ITER,IER)
      IF(IER.NE.0)RETURN
C CALCULATE DIMENSIONLESS SHEAR STRESS AND FLOW AND TORQUE PARAMETERS
      FLO=RG(1,1)*H(1,1)*V(1,1)*2.D0*PI
      CALL TORQ(NOI,IFACE,IDIR,RE,REC,P1R,OMT,NREG,NRSUB,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,UHG,VHG,
     +RG,ZTG,H,U,V,TAU,TOR)
C CALCULATE LOAD UNDER FACE SEAL
      IF(IFACE.EQ.1)CALL FORCE(NREG,NRSUB,RG,ZTG,P,W)
C CALCULATE 0 FREQUENCY STIFFNESS, CK,  AND DAMPING
      CALL OUTSCR('    SECOND ORDER SOLUTION',0)
      CALL KBCAL(NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,OMT,0.D0,DUT,ZET,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,CK,CB,IER)
      IF(IER.NE.0)RETURN
      IF(ABS(OMDT).GT.1.D-8)THEN
C IF NON 0 VALUE OF DISTURBANCE FREQ. AM AND CB WILL CONTAIN STIFFNESS AND
```



```
C DAMPING AT DISTURBANCE FREQ.
      IAMASS=0
      CALL KBCAL(NOI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,OMT,OMDT,DUT,ZET,
     +  IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,AM,CB,IER)
      IF(IER.NE.0)RETURN
      ELSE
C OTHERWISE AM WILL CONTAIN MASS MATRIX AND CB WILL BE 0 FREQ DAMPING
      IAMASS=1
      CALL KBCAL(NOI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,OMT,1.D0,DUT,ZET,
     +  IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,AM,TMP,IER)
      IF(IER.NE.0)RETURN
      DO 8 I=1,4-IFACE
      DO 8 J=1,4-IFACE
  8   AM(I,J)=CK(I,J)-AM(I,J)
      ENDIF
      RETURN
      END
```

```
      FUNCTION FLMSHP(X)
C THIS IS THE USER DEFINED FILM SHAPE FUNCTION
C X IS THE DISTANCE FROM THE CENTER OF THE SEAL DIVIDED BY THE SEALING
C LENGTH , L . -.5 <= X <= .5
C FOR A SHAFT SEAL X = S/(2*L/D)   (S IS ZTG IN CODE)
C FOR A FACE SEAL X=(S+L/D-1)/(2*L/D)
C FLMSHP IS THE SHAPE OF THE FILM (DIMENSIONAL)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BFSHP/HTAP,HBRL
      FLMSHP=-HTAP*X+HBRL*(1.DO-(2.DO*X)**2)
      RETURN
      END
```



```
      FUNCTION FA(RE,H)
C USER DEFINABLE FRICTION FACTOR FOR MOVING SURFACE
C CALLED BY RECAL,TORQ,PHIPSI
C COMMON BLOCK PASSED FROM MAIN PROGRAM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BFAFB/EMA,ENA,EMB,ENB
      FA=MAX(24.DO/RE,ENA*RE**EMA)
C H IS NOT USED NOW BUT MAY BE IN FUTURE FOR TREATING ROUGHNESS
      H1=H
      RETURN
      END
```

```
      FUNCTION FB(RE,H)
C USER DEFINABLE FRICTION FACTOR FOR STATIONARY SURFACE
C CALLED BY RECAL,TORQ,PHIPSI
C COMMON BLOCK PASSED FROM MAIN PROGRAM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON/BFAFB/EMA,ENA,EMB,ENB
      FB=MAX(24.DO/RE,ENB*RE**EMB)
C H IS NOT USED NOW BUT MAY BE IN FUTURE FOR TREATING ROUGHNESS
      H1=H
      RETURN
      END
```



```
      FUNCTION DELTP(RE,V,H,HSTEP,ZET)
C USER DEFINABLE FUNCTION FOR COMPUTING LOSS COEFFICIENTS
C PRESSURE CHANGE (DOWNSREAM -UPSTREAM) DUE TO SUDDEN CHANGE IN CROSS SECTION
C HSTEP = STEP HEIGHT (H_UPSTREAM-H_DOWNSTREAM)
C CALLED BY UVPCAL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IF(ABS(HSTEP).LT.1.D-8)THEN
        DELTP=0.DO
        GO TO 99
        ELSE IF(HSTEP.LT.0.DO)THEN
C COMPUTE LOSS COEFFICIENT FOR EXPANSION
        ZET1=(1.DO-H/(H+HSTEP))**2
        ELSE
C USE INPUT LOSS COEFFICIENT FOR CONTRACTION (ZET) OR COMPUTE IT FROM RE
        RE1=RE*H*ABS(V)
        ZET1=ZET
      ENDIF
      DELTP=-(1.DO+ZET1)
      IF(HSTEP.LT.1.D8)DELTP=DELTP+(H/(H+HSTEP))**2
      DELTP=DELTP*V*V
   99 RETURN
      END
```

```
      FUNCTION DIRFCN(RE,REC,P1R,OMT,VCON,IFACE,ID,R,H,U,DV)
C CALCULATES PRIMARY FLOW DERIVATIVES FOR TAN VEL. U (ID=1)
C OR PRESSURE P (ID=2)
C CALLED BY UVPCAL
C CALLS PHIPSI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      V=VCON/R/H
      CALL PHIPSI(IFACE,RE,REC,OMT,R,H,U,V,PHI,PSI)
      IF(ID.EQ.1)DIRFCN=-PHI/REC/V
      IF(ID.EQ.2)DIRFCN=-P1R*(PSI+REC*V*DV)
      RETURN
      END
```



```
      SUBROUTINE RZGRID(IFACE,ELT,NREG,NRSUB,ELFR,RG,ZTG)
C GENERATES R (RZ) AND Z (ZTG) GRIDS
C TO LOCATE STARTING POINT FROM Z ORIGIN ADD INSIDE RADIUS
C FOR FACE SEAL (IFACE=1), SUBTRACT L/D FOR SHAFT SEAL
C R = Z FOR FACE SEAL AND R=1 FOR SHAFT SEAL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZTG(NDZ,NDREG),ELFR(NDREG)
      IF(IFACE.NE.1)IFACE=0
      ZTG(1,1)=-ELT
      RG(1,1)=1.D0
      IF(IFACE.EQ.1)THEN
        ZTG(1,1)=1.D0-2.D0*ELT
        RG(1,1)=ZTG(1,1)
      ENDIF
      DO 102 KK=1,NREG
        IF(KK.GT.1)THEN
          ZTG(1,KK)=ZTG(NRS+1,KK-1)
          RG(1,KK)=RG(NRS+1,KK-1)
        ENDIF
        NRS=NRSUB(KK)
        DZF=2.D0*ELFR(KK)*ELT/NRS
        DO 102 JJ=1,NRS
          ZTG(JJ+1,KK)=ZTG(JJ,KK)+DZF
          RG(JJ+1,KK)=1.D0
          IF(IFACE.EQ.1)RG(JJ+1,KK)=ZTG(JJ+1,KK)
 102  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE VISOLV(TOLV,NITV,VI,NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,ITER,IER)
C SOLVES FOR INLET VELOCITY VI USING NEWTONS METHOD
C ON INPUT VI IS INITIAL GUESS. IER=2 IF NOT CONVERGED.
C CALLED BY TSEAL
C CALLS UVPCAL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION ALP(NDREG),SBET(NDREG),CBET(NDREG),DELT(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
      DO 5 J=1,NITV
        ITER=J
        CALL UVPCAL(VI+DUT,NOI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +  IGROT,0,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,P,F2,IER)
        CALL UVPCAL(VI,NOI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +  IGROT,1,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,P,F1,IER)
        DV=F1*DUT/(F2-F1)
        VI=VI-DV
        VI1=ABS(VI)+TOLV
        IF(ABS(DV)/VI1.LT.TOLV)GO TO 6
  5   CONTINUE
      IF(NITV.GT.1)IER=2
  6   CALL UVPCAL(VI,NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +IGROT,1,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,F1,IER)
      RETURN
      END
```



```
      SUBROUTINE UVPCAL(VI,NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +IGROT,IUHG,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,PEXIT,IER)
C GENERATES PRESSURE (P) AND VELOCITIES (U,V) BASED ON INITIAL VALUE OF
C V (VI)
C CALLED BY VISOLV
C CALLS UVPIN WHEN REC>0 OR UVPNOI WHEN REC = 0
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION ALP(NDREG),SBET(NDREG),CBET(NDREG),DELT(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
      IF(NOI.EQ.1)THEN
        CALL UVPNOI(VI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,PIN,OMT,DUT,
     +  IGROT,IUHG,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,P,PEXIT,IER)
        ELSE
        CALL UVPIN(VI,IFACE,IDIR,NREG,NRSUB,
     +  RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +  IGROT,IUHG,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +  RG,ZTG,H,U,V,P,PEXIT,IER)
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE UVPNOI(VI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,DUT,
     +IGROT,IUHG,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,PEXIT,IER)
C GENERATES PRESSURE (P) AND VELOCITIES (U,V) BASED ON INITIAL VALUE OF
C V (VI) WITHOUT INERTIA EFFECTS
C CALLED UVPCAL
C CALLS DIRFCN,USOLV (NO GROOVES) OR PHIPSG (FOR SPIRAL GROOVES)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
C COMMON BLOCK USED LOCALLY IN THIS ROUTINE
      COMMON/BGRLCL/UHL(NDZ,NDREG),VHL(NDZ,NDREG)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION ALP(NDREG),SBET(NDREG),CBET(NDREG),DELT(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
C SET UP STARTING CONDITIONS AND LOOPING PARAMETERS FOR PRIMARY FLOW SOLUTION
      KST=1
      KEN=NREG
      IF(IDIR.EQ.-1)THEN
        KST=NREG
        KEN=1
      ENDIF
      DO 30 K=KST,KEN,IDIR
        JST=1
        JEN=NRSUB(K)
        IF(IDIR.EQ.-1)THEN
          JST=JEN+1
          JEN=2
        ENDIF
        IF(K.EQ.KST)THEN
          VCON=VI*RG(JST,KST)*H(JST,KST)*IDIR
          V(JST,KST)=VI*IDIR
          P(JST,KST)=PIN
          ELSE
          V(JST,K)=VCON/RG(JST,K)/H(JST,K)
          P(JST,K)=P(J1,K1)
        ENDIF
        IF(IGROT(K).EQ.-1)THEN
          CALL USOLV(RE,OMT,RG(JST,K),H(JST,K),U(JST,K),
     +    V(JST,K),DUT,IUHG,IER)
          ELSE
          CALL PHIPSG(1,IFACE,IUHG,RE,REC,OMT,RG(JST,K),H(JST,K),
     +    U(JST,K),V(JST,K),UHL(JST,K),VHL(JST,K),
     +    ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +    IGROT(K),PHI,PSI,IER)
         ENDIF
         IF(IER.NE.0)RETURN
        K1=K
        DO 30 J=JST,JEN,IDIR
          J1=J+IDIR
          RB=.5D0*(RG(J1,K)+RG(J,K))
          HB=.5D0*(H(J1,K)+H(J,K))
          DX=ZTG(J1,K)-ZTG(J,K)
          V(J1,K)=VCON/RG(J1,K)/H(J1,K)
          IF(IGROT(K).EQ.-1)THEN
           CALL USOLV(RE,OMT,RG(J1,K),H(J1,K),U(J1,K),V(J1,K),
     +      DUT,IUHG,IER)
           IF(IER.NE.0)RETURN
           P(J1,K)=P(J,K)+DX*DIRFCN(RE,REC,P1R,OMT,VCON,IFACE,2,
     +             RB,HB,.5D0*(U(J1,K)+U(J,K)),0.D0)
           ELSE
           VB=VCON/RB/HB
           CALL PHIPSG(1,IFACE,IUHG,RE,REC,OMT,RG(J1,K),H(J1,K),
     +     U(J1,K),V(J1,K),UHL(J1,K),VHL(J1,K),
```



```
     +      ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +      IGROT(K),PHI,PSI,IER)
           IF(IER.NE.0)RETURN
           CALL PHIPSG(0,IFACE,IUHG,RE,REC,OMT,RB,HB,
     +      .5D0*(U(J1,K)+U(J,K)),VB,UHG(J,K),VHG(J,K),
     +      ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +      IGROT(K),PHI,PSI,IER)
           P(J1,K)=P(J,K)-DX*P1R*PSI
           IF(IER.NE.0)RETURN
         ENDIF
 30   CONTINUE
      PEXIT=P(J1,K1)
      RETURN
      END
```

```
      SUBROUTINE USOLV(RE,OMT,R,H,U,V,DUT,ILAST,IER)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C SOLVES FOR EQUILIBRIUM TANGENTIAL VELOCITY WHEN THERE ARE NO GROOVES
C CALLED BY UVPNOI
C CALLS PHIPSI
      IF(ABS(OMT).LT.DUT)THEN
        U=0.DO
        RETURN
      ENDIF
      DU=DUT*OMT
      TOL=100.DO*ABS(DU)
      IF(ILAST.EQ.0)U=.5DO*OMT
      DO 5 I=1,30
        CALL PHIPSI(O,RE,0.DO,OMT,R,H,U,V,PHI,PSI)
        CALL PHIPSI(O,RE,0.DO,OMT,R,H,U+DU,V,DPHI,PSI)
        DLT=PHI*DU/(DPHI-PHI)
        U=U-DLT
        IF(ABS(DLT).LT.TOL)GO TO 6
    5 CONTINUE
      IER=2
    6 RETURN
      END
```



```
      SUBROUTINE UVPIN(VI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,PIN,OMT,OMOT,DUT,ZET,
     +IGROT,IUHG,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,P,PEXIT,IER)
C GENERATES PRESSURE (P) AND VELOCITIES (U,V) BASED ON INITIAL VALUE OF
C V (VI) WHEN INERTIA IS PRESENT
C CALLED BY UVPCAL
C CALLS DIRFCN,DELTP,PHIPSG (FOR SPIRAL GROOVES)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
C COMMON BLOCK USED LOCALLY IN THIS ROUTINE
      COMMON/BGRLCL/UHL(NDZ,NDREG),VHL(NDZ,NDREG)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG),P(NDZ,NDREG)
      DIMENSION ALP(NDREG),SBET(NDREG),CBET(NDREG),DELT(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
C SET UP STARTING CONDITIONS AND LOOPING PARAMETERS FOR PRIMARY FLOW SOLUTION
      KST=1
      KEN=NREG
      RV20=.5DO*REC*P1R
      IF(IDIR.EQ.-1)THEN
        KST=NREG
        KEN=1
      ENDIF
      DO 30 K=KST,KEN,IDIR
        JST=1
        JEN=NRSUB(K)
        IF(IDIR.EQ.-1)THEN
          JST=JEN+1
          JEN=2
        ENDIF
        IF(K.EQ.KST)THEN
          VCON=VI*RG(JST,KST)*H(JST,KST)*IDIR
          V(JST,KST)=VI*IDIR
          U(JST,KST)=OMOT*RG(JST,KST)
          P(JST,KST)=PIN+
     +    RV20*DELTP(RE,V(JST,KST),H(JST,KST),1.D10,ZET(KST))
          ELSE
          V(JST,K)=VCON/RG(JST,K)/H(JST,K)
          U(JST,K)=U(J1,K1)
          P(JST,K)=P(J1,K1)+
     +    RV20*DELTP(RE,V(JST,K),H(JST,K),H(J1,K1)-H(JST,K),ZET(K))
        ENDIF
        K1=K
        DO 30 J=JST,JEN,IDIR
          J1=J+IDIR
          RB=.5DO*(RG(J1,K)+RG(J,K))
          HB=.5DO*(H(J1,K)+H(J,K))
          DX=ZTG(J1,K)-ZTG(J,K)
          V(J1,K)=VCON/RG(J1,K)/H(J1,K)
          IF(IGROT(K).EQ.-1)THEN
            U1=DIRFCN(RE,REC,P1R,OMT,VCON,IFACE,1,
     +         RB,HB,U(J,K),0.DO)
            DU1=(DIRFCN(RE,REC,P1R,OMT,VCON,IFACE,1,
     +           RB,HB,U(J,K)+DUT,0.DO)-U1)/DUT
            U(J1,K)=U(J,K)+DX*U1/(1.DO-.5DO*DX*DU1)
            P(J1,K)=P(J,K)+DX*DIRFCN(RE,REC,P1R,OMT,VCON,IFACE,2,
     +                RB,HB,.5DO*(U(J1,K)+U(J,K)),(V(J1,K)-V(J,K))/DX)
          ELSE
          VB=VCON/RB/HB
          CALL PHIPSG(O,IFACE,IUHG,RE,REC,OMT,RB,HB,
     +    U(J,K),VB,UHL(J,K),VHL(J,K),
     +    ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +    IGROT(K),PHI,PSI,IER)
          IF(IER.NE.O)RETURN
```

```
          U1=-PHI/REC/V(J,K)
          CALL PHIPSG(0,IFACE,-1,RE,REC,OMT,RB,HB,
     +    U(J,K)+DUT,VB,UHL(J,K),VHL(J,K),
     +    ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +    IGROT(K),PHI,PSI,IER)
          IF(IER.NE.0)RETURN
          DU1=(-PHI/REC/VB-U1)/DUT
          U(J1,K)=U(J,K)+DX*U1/(1.D0-.5D0*DX*DU1)
          CALL PHIPSG(0,IFACE,IUHG,RE,REC,OMT,RB,HB,
     +    .5D0*(U(J1,K)+U(J,K)),VB,UHG(J,K),VHG(J,K),
     +    ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),
     +    IGROT(K),PHI,PSI,IER)
          P(J1,K)=P(J,K)-DX*P1R*(PSI+REC*VB*(V(J1,K)-V(J,K))/DX)
          IF(IER.NE.0)RETURN
        ENDIF
 30   CONTINUE
      PEXIT=P(J1,K1)
      RETURN
      END
```



```
      SUBROUTINE RECAL(REL,RE,FBAR,NITV,TOLV,IER)
C USES NEWTON ITERATION TO GET CHARACTERISTIC REYNOLDS NUMBER FOR TURBULENT
C POISEUILLE FLOW WITH UNIFORM CLEARANCE
C REL = LAMINAR REYNOLDS NUMBER (INPUT)
C RE  = REYNOLDS NUMBER
C IER = ERROR CODE, 0 IF OK
C CALLS FA,FB
C CALLED BY MAIN PROGRAM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      IER=0
      RE=REL
      DRE=1.D-6*RE
      DO 5 I=1,NITV
        I1=I
        FBAR=(FA(RE,1.D0)+FB(RE,1.D0))/2.D0
        DFBAR=((FA(RE+DRE,1.D0)+FB(RE+DRE,1.D0))/2.D0-FBAR)/DRE
        DELT=(RE*RE*FBAR-24.D0*REL)/(RE*RE*DFBAR+2.D0*RE*FBAR)
        RE=RE-DELT
        IF(ABS(DELT/RE).LT.TOLV)GO TO 6
 5    CONTINUE
      IER=1
 6    RETURN
      END
```

```
      SUBROUTINE FORCE(NREG,NRSUB,RG,ZTG,P,W)
C COMPUTES DIMENSIONLESS LOAD, W, FROM PRIMARY PRESSURE DISTRIBUTION
C ONLY MEANINGFUL FOR FACE SEAL
C CALLED BY TSEAL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZTG(NDZ,NDREG),P(NDZ,NDREG)
      W=0.D0
      DO 5 K=1,NREG
      NRS=NRSUB(K)
      DO 5 J=1,NRS
      J1=J+1
    5 W=W+(P(J,K)+P(J1,K))*(RG(J,K)+RG(J1,K))*(ZTG(J1,K)-ZTG(J,K))/4.D0
      W=W*8.D0*ATAN(1.D0)
      RETURN
      END
```



```
      SUBROUTINE TORQ(NOI,IFACE,IDIR,RE,REC,P1R,OMT,NREG,NRSUB,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,UHG,VHG,
     +RG,ZTG,H,U,V,TAU,TOR)
C CALCULATES SHEAR STRESS ON MOVING SURFACE AND TORQUE INTEGRAL
C SHEAR STRESSES AT ARE AT HALF GRID POINTS
C CALLED BY TSEAL
C CALLS FA AND FB
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZTG(NDZ,NDREG),TAU(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG)
      DIMENSION ALP(NDREG),DELT(NDREG),SBET(NDREG),CBET(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION ENGP(NDREG)
C REDUCE COUETTE PART OF SHEAR STRESS BY 3 FOR LAMINAR FLOW
C OR BY 1.2 AS SUGGESTED BY HIRS FOR TURBULENT FLOW
      DATA DLAM,DHIRS/3.D0,1.2D0/
      VCON=V(1,1)*RG(1,1)*H(1,1)
      TOR=0.D0
C NEED TO SET INTEGRATION DIRECTION TO PROPERLY GET UHG AND VHG AT HALF GRID
      KST=1
      KEN=NREG
      IF(IDIR.EQ.-1)THEN
        KST=NREG
        KEN=1
      ENDIF
      DO 4 K=KST,KEN,IDIR
        JST=1
        JEN=NRSUB(K)
        IF(IDIR.EQ.-1)THEN
          JST=JEN+1
          JEN=2
        ENDIF
        DO 4 J=JST,JEN,IDIR
          J1=J+IDIR
          HBAR=.5D0*(H(J,K)+H(J1,K))
          UBAR=.5D0*(U(J,K)+U(J1,K))
          RBAR=.5D0*(RG(J,K)+RG(J1,K))
          VBAR=VCON/RBAR/HBAR
          DPCOR=0.D0
          IF(IGROT(K).EQ.-1)THEN
            RA=RE*HBAR*SQRT((UBAR-RBAR*OMT)**2+VBAR**2)
            RAFA=RA*FA(RA,HBAR)
C REDUCE SHEAR STRESS BY FACTOR OF 3 FOR LAMINAR FLOW
            DLM=DHIRS
            IF(ABS(RAFA-24.D0).LT.1.D-10)DLM=DLAM
            TAU(J,K)=P1R*RAFA*(UBAR-RBAR*OMT)/HBAR/DLM
            ELSE
            HR=HBAR-ALP(K)*DELT(K)
            HG=HR+DELT(K)
            UHR=(UBAR*HBAR-UHG(J,K)*ALP(K))/(1.D0-ALP(K))
            VHR=(VBAR*HBAR-VHG(J,K)*ALP(K))/(1.D0-ALP(K))
            RAG=RE*SQRT((UHG(J,K)-RBAR*OMT*HG)**2+VHG(J,K)**2)
            RAR=RE*SQRT((UHR-RBAR*OMT*HR)**2+VHR**2)
            RAFAG=RAG*FA(RAG,HG)
            RAFAR=RAR*FA(RAR,HR)
            RBG=RE*SQRT(UHG(J,K)**2+VHG(J,K)**2)
            RBR=RE*SQRT(UHR**2+VHR**2)
            RBFBG=RBG*FB(RBG,HG)
            RBFBR=RBR*FB(RBR,HR)
            TAUGA=RAFAG*(UHG(J,K)-RBAR*OMT*HG)/HG**2
            TAUGB=-RBFBG*UHG(J,K)/HG**2
            TAURA=RAFAR*(UHR-RBAR*OMT*HR)/HR**2
            TAURB=-RBFBR*UHR/HR**2
C SET COUETTE REDUCTION FACTOR
            DLM=DHIRS
```

```
            IF(ABS(RAFAR-24.D0).LT.1.D-10.OR.ABS(RAFAG-24.D0).LT.1.D-10)
     +      DLM=DLAM
C ATTEMPT TO SPLIT OFF COUETTE AND POISEUILLE PORTIONS OF SHEAR STRESS
            TAURC=(TAURA+TAURB)/2.D0/DLM
            TAUGC=(TAUGA+TAUGB)/2.D0/DLM
            TAURP=(TAURA-TAURB)/2.D0
            TAUGP=(TAUGA-TAUGB)/2.D0
C IF GROOVES ROTATE CORRECT FOR FORCES AT GROOVE EDGES
            IF(IGROT(K).EQ.1)THEN
              TAUGP=TAUGP*(1.D0-2.D0*DELT(K)/HG)
C GET EFFECTS OF LOCAL INERTIA DROP, DPCOR
              QN=HBAR*(UBAR-RBAR*OMT)*SBET(K)
              IF(NOI.NE.1)QN=QN-HBAR*VBAR*CBET(K)
              DPCOR=REC*ENGP(K)/RBAR*DELTP(RE,QN/HG,HG,-DELT(K),0.D0)*
     +              SIGN(1.D0,SBET(K)*QN)
C
              IF(NOI.NE.1)THEN
                TICOR=(U(J1,K)-U(J,K))/(ZTG(J1,K)-ZTG(J,K))
                IF(IFACE.EQ.1)TICOR=TICOR+UBAR/RBAR
                TAUGP=TAUGP-DELT(K)*REC*VBAR*TICOR
              ENDIF
            ENDIF
            TAU(J,K)=P1R*(ALP(K)*(TAUGC+TAUGP)+
     +              (1.D0-ALP(K))*(TAURP+TAURC)+DPCOR)
          ENDIF
          TOR=TOR+TAU(J,K)*RBAR**2*(ZTG(J1,K)-ZTG(J,K))*IDIR
   4  CONTINUE
C MULTIPLY BY 2 PI AND CHANGE SIGN SO THAT TORQUE IS + WHEN IT OPPOSES MOTION
      TOR=-TOR*8.D0*ATAN(1.D0)
      RETURN
      END
```



```
      SUBROUTINE PHIPSG(NOI,IFACE,IUHG,RE,REC,OMT,R,H,U,V,UHG,VHG,
     +ALP,S,C,DELT,ENGP,ZETG,IGROT,PHI,PSI,IER)
C GENERATES GLOBAL TURBULENCE FUNCTIONS FOR SPIRAL GROOVES
C CALLED BY UVPNOI,UVPIN,DSOLV
C CALLS PHIPSQ,MATINV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Q(4),B(4),INDEX(4,3),A(4,4),D(4,4),E(4)
      DATA A/16*0.D0/
      DQ=1.D-6
      HR=H-ALP*DELT
      HG=HR+DELT
      UH=U*H
      VH=V*H
      IF(IUHG.EQ.0)THEN
        IF(NOI.EQ.1)UH=.5D0*H*R*OMT
        Q(1)=UH
        Q(2)=VH
        ELSE
        Q(1)=UHG
        Q(2)=VHG
      ENDIF
      Q(3)=(UH-Q(1)*ALP)/(1.D0-ALP)
      Q(4)=(VH-Q(2)*ALP)/(1.D0-ALP)
      DO 30 L=1,30
        CALL PHIPSQ(RE,OMT,R,HG,Q(1),Q(2),B(1),B(2))
        CALL PHIPSQ(RE,OMT,R,HG,Q(1)+DQ,Q(2),A(1,1),A(2,1))
        CALL PHIPSQ(RE,OMT,R,HG,Q(1),Q(2)+DQ,A(1,2),A(2,2))
        CALL PHIPSQ(RE,OMT,R,HR,Q(3),Q(4),B(3),B(4))
        CALL PHIPSQ(RE,OMT,R,HR,Q(3)+DQ,Q(4),A(3,3),A(4,3))
        CALL PHIPSQ(RE,OMT,R,HR,Q(3),Q(4)+DQ,A(3,4),A(4,4))
        DO 8 K1=1,2
          K=2*(K1-1)
          DO 8 I=1,2
            DO 7 J=1,2
  7         A(I+K,J+K)=(A(I+K,J+K)-B(I+K))/DQ
            B(I+K)=-B(I+K)
            DO 8 J=1,2
  8     B(I+K)=B(I+K)+A(I+K,J+K)*Q(J+K)
        E(1)=C*B(1)+S*B(2)-C*B(3)-S*B(4)
        DO 9 J=1,4
  9     D(1,J)=C*A(1,J)+S*A(2,J)-C*A(3,J)-S*A(4,J)
        D(2,1)=S
        D(2,2)=-C
        D(2,3)=-S
        D(2,4)=C
        E(2)=R*OMT*DELT*S*IGROT
        IF(NOI.EQ.1)THEN
          E(3)=ALP*B(1)+(1.D0-ALP)*B(3)
          DO 10 J=1,4
 10       D(3,J)=ALP*A(1,J)+(1.D0-ALP)*A(3,J)
C         IF(IFACE.EQ.1)THEN
C           COR=REC*VH/R/H**2
C           D(3,1)=D(3,1)-ALP*COR
C           D(3,3)=D(3,3)-(1.D0-ALP)*COR
C         ENDIF
          ELSE
          D(3,1)=ALP
          D(3,2)=0.D0
          D(3,3)=1.D0-ALP
          D(3,4)=0.D0
          E(3)=UH
        ENDIF
        D(4,1)=0.D0
        D(4,2)=ALP
        D(4,3)=0.D0
        D(4,4)=1.D0-ALP
```

```
        E(4)=VH
        CALL MATINV(D,E,DETER,4,1,ID,4,INDEX)
        IF(ID.NE.1)GO TO 99
        ICNV=0
        EMX=-1.D20
        DO 11 I=1,4
          EMX=MAX(EMX,ABS(E(I)-Q(I)))
  11    Q(I)=E(I)
        IF(EMX.LT.1.D-4)GO TO 31
  30  CONTINUE
      GO TO 99
  31  CALL PHIPSQ(RE,OMT,R,HG,Q(1),Q(2),PHIG,PSIG)
      CALL PHIPSQ(RE,OMT,R,HR,Q(3),Q(4),PHIR,PSIR)
      IF(NOI.EQ.1)U=(ALP*Q(1)+(1.DO-ALP)*Q(3))/H
      IF(IUHG.NE.-1)THEN
        UHG=Q(1)
        VHG=Q(2)
      ENDIF
      ERR=C*PHIG+S*PSIG-C*PHIR-S*PSIR
      IF(ERR.GT.1.D-4)IER=4
      PHI=ALP*PHIG+(1.DO-ALP)*PHIR
      PSI=ALP*PSIG+(1.DO-ALP)*PSIR
C ADD EFFECTS OF LOCAL INERTIA DROP
C   NORMAL FLOW
      QN=H*(U-R*OMT*IGROT)*S
      IF(NOI.NE.1)QN=QN-H*V*C
C     CONTRACTION LOSS COEFF, ZETG
      PDRP=DELTP(RE,QN/HG,HG,HR-HG,0.DO)+DELTP(RE,QN/HR,HR,HG-HR,ZETG)
      PHI=PHI-SIGN(1.DO,S*QN)*REC*ENGP/R*PDRP
      PSI=PSI+SIGN(1.DO,C*QN)*REC*ENGP/R*PDRP*ABS(C/S)
      IF(IFACE.NE.1)RETURN
      IF(NOI.NE.1)PHI=PHI+REC*U*V/R
      PSI=PSI-REC*U*U/R
      RETURN
  99  IER=4
      RETURN
      END
```



```
      SUBROUTINE PHIPSQ(RE,OMT,R,H,QT,QS,PHI,PSI)
C GENERATES TURBULENCE FUNCTIONS PHI,PSI BASED ON FLOW RATHER THAN VELOCITY
C EXCLUDES CENTRIFUGAL AND CORIOLIS TERMS FOR FACE SEAL
C CALLS PHIPSI
C CALLED BY PHIPSG
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CALL PHIPSI(O,RE,0.DO,OMT,R,H,QT/H,QS/H,PHI,PSI)
      RETURN
      END
```

```
      SUBROUTINE PHIPSI(IFACE,RE,REC,OMT,R,H,U,V,PHI,PSI)
C GENERATES TURBULENCE FUNCTIONS PHI,PSI
C INCLUDES CENTRIFUGAL AND CORIOLIS TERMS FOR FACE SEAL
C CALLS FA,FB
C CALLED BY DIRFCN,DSOLV,PHIPSQ
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      RA=RE*H*SQRT((U-R*OMT)**2+V*V)
      RB=RE*H*SQRT(U*U+V*V)
      CK2=RA*FA(RA,H)
      CK1=CK2+RB*FB(RB,H)
      PHI=(CK1*U-CK2*R*OMT)/H**2
      PSI=CK1*V/H**2
      IF(IFACE.NE.1)RETURN
      PHI=PHI+REC*U*V/R
      PSI=PSI-REC*U*U/R
      RETURN
      END
```



```
      SUBROUTINE DSOLV(NOI,IFACE,RE,REC1,P1R,OMT,OMDT,DUT,
     +ALP,SBET,CBET,DELT,ENGP,ZETG,IGROT,
     +R1,Z1,H1,U1,V1,UHG,VHG,R,Z,H,U,V,Y,FOI,EMI,IER)
C UPDATES Y(I,L,N) TO NEXT Z POSITION
C GENERATES MATRICES FOR DISURBANCE EQUATIONS AT ONE VALUE OF Z
C DISTURBANCE EQUATIONS ARE IN FORM {DY/DZ} = [A]{Y} + {B}
C WHERE I=1,2,3 CORRESPONDS TO P,V,U DISTURBANCES
C L=1,2,3 CORRESPONDS TO COMPLIMENTARY SOLUTION,TILT,RADIAL DISP RESP. WHEN N<3
C N=1,2,3 CORRESPONDS TO EXP(I*(THETA+OMDT*T)),EXP(I*(THETA-OMDT*T)),
C                        EXP(I*OMDT*T) RESP.
C N=3 IS FOR AXIAL DISTURBANCE APPLIED TO FACE SEAL FOR WHICH CASE
C     L=1,2 CORRESPONDS TO COMP. SOL. AND AXIAL DISP RESP.
C CALLS PHIPSI,PHIPSG,ESET,CMATIN
C CALLED BY KBCAL
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMPLEX*16 A(3,3),B(3,3),TI(3),OI(3),UR,DETER,Y(3,3,3),
     +FOI(3,3),EMI(3,3),TMP,DFI
      DIMENSION E(3),DE(3),LABEL(3,3)
      DATA TI,OI/(0.D0,1.D0),(0.D0,1.D0),(0.D0,0.D0),
     +          (0.D0,1.D0),(0.D0,-1.D0),(0.D0,1.D0)/
      REC=REC1
      IF(NOI.EQ.1)REC=0.D0
      UB=.5D0*(U+U1)
      HB=.5D0*(H+H1)
      RB=.5D0*(R+R1)
      VB=V*R*H/RB/HB
      ZB=.5D0*(Z+Z1)
      DZ=Z-Z1
      DZ2=.5D0*DZ
      DU=(U-U1)/DZ
      DV=(V-V1)/DZ
      DRH=(RB*(H-H1)+HB*(R-R1))/DZ
      REP=P1R*REC
C CALCULATE TURBULENT FUNCTIONS AND THEIR DERIVATIVES
      IF(IGROT.EQ.-1)THEN
        CALL PHIPSI(IFACE,RE,REC,OMT,RB,HB,UB,VB,PHI,PSI)
        CALL PHIPSI(IFACE,RE,REC,OMT,RB,HB+DUT,UB,VB,PHIH,PSIH)
        CALL PHIPSI(IFACE,RE,REC,OMT,RB,HB,UB+DUT,VB,PHIU,PSIU)
        CALL PHIPSI(IFACE,RE,REC,OMT,RB,HB,UB,VB+DUT,PHIV,PSIV)
        ELSE
        UHGB=UHG
        VHGB=VHG
        CALL PHIPSG(0,IFACE,1,RE,REC,OMT,RB,HB,UB,VB,UHGB,VHGB,
     +  ALP,SBET,CBET,DELT,ENGP,ZETG,IGROT,PHI,PSI,IER)
        IF(IER.NE.0)RETURN
        CALL PHIPSG(0,IFACE,-1,RE,REC,OMT,RB,HB+DUT,UB,VB,UHGB,VHGB,
     +  ALP,SBET,CBET,DELT,ENGP,ZETG,IGROT,PHIH,PSIH,IER)
        IF(IER.NE.0)RETURN
        CALL PHIPSG(0,IFACE,-1,RE,REC,OMT,RB,HB,UB+DUT,VB,UHGB,VHGB,
     +  ALP,SBET,CBET,DELT,ENGP,ZETG,IGROT,PHIU,PSIU,IER)
        IF(IER.NE.0)RETURN
        CALL PHIPSG(0,IFACE,-1,RE,REC,OMT,RB,HB,UB,VB+DUT,UHGB,VHGB,
     +  ALP,SBET,CBET,DELT,ENGP,ZETG,IGROT,PHIV,PSIV,IER)
        IF(IER.NE.0)RETURN
      ENDIF
      PHIH=(PHIH-PHI)/DUT
      PSIH=(PSIH-PSI)/DUT
      PHIU=(PHIU-PHI)/DUT
      PSIU=(PSIU-PSI)/DUT
      PHIV=(PHIV-PHI)/DUT
      PSIV=(PSIV-PSI)/DUT
      IF(IFACE.EQ.1.AND.NOI.EQ.1)PSIU=PSIU-REC1/RB*2.D0*UB
      NMAX=2+IFACE
      LMAX=3-IFACE
      DO 5 N=1,NMAX
C SET DISPLACEMENT AMPLITUDES
```

```
        CALL ESET(ZB,N,E,DE)
        UR=OMDT*OI(N)+UB/RB*TI(N)
        A(1,1)=(0.D0,0.D0)
        A(1,2)=(REC*(DV-VB/RB/HB*DRH)+PSIV+UR*REC1)*P1R
        A(1,3)=P1R*PSIU-TI(N)*REP*VB/RB
        A(2,1)=(0.D0,0.D0)
        A(2,2)=DCMPLX(DRH/RB/HB,0.D0)
        A(2,3)=TI(N)/RB
        A(3,1)=TI(N)/RB/P1R
        A(3,2)=DCMPLX(DU*REC+PHIV,0.D0)
        A(3,3)=(PHIU+UR*REC1)
C L=1 IS COMPLEMENTARY SOLUTION
        DO 7 I=1,3
    7   B(I,1)=(0.D0,0.D0)
        DO 6 L=2,LMAX
          TMP=(VB*(RB*DE(L)+IFACE*E(L))+RB*E(L)*(DV+UR))/RB/HB
          B(1,L)=P1R*PSIH*E(L)-REP*VB*TMP
          B(2,L)=TMP
          B(3,L)=DCMPLX(E(L)*PHIH,0.D0)
    6   CONTINUE
C IF AXIAL (RADIAL) INERTIA IS INCLUDED DIVIDE U EQUATION BY COEFF OF DU/DS
        IF(NOI.NE.1)THEN
          NEQ=3
          DO 20 J=1,3
   20     A(3,J)=A(3,J)/REC/VB
          DO 21 L=1,LMAX
   21     B(3,L)=B(3,L)/REC/VB
          ELSE
          NEQ=2
          DO 23 I=1,2
            A(I,3)=A(I,3)/A(3,3)
            DO 24 L=1,LMAX
   24       B(I,L)=B(I,L)-A(I,3)*B(3,L)
            DO 23 J=1,2
   23     A(I,J)=A(I,J)-A(I,3)*A(3,J)
        ENDIF
C REPLACE {B} WITH {B}-[A]{Y}
        DO 22 L=1,LMAX
          DO 22 I=1,NEQ
          DO 22 J=1,NEQ
   22   B(I,L)=B(I,L)-A(I,J)*Y(J,L,N)
C REPLACE [A] WITH [I]+DZ/2*[A]
        DO 8 I=1,NEQ
        DO 8 J=1,NEQ
          A(I,J)=DZ2*A(I,J)
          IF(I.EQ.J)A(I,J)=1.D0+A(I,J)
    8   CONTINUE
C SOLVE EQUATIONS FOR ALL LMAX RIGHT HAND SIDE VECTORS IN ONE SHOT
        CALL CMATIN(A,B,DETER,NEQ,LMAX,ID,3,LABEL)
        IF(ID.NE.1)THEN
          IER=3
          RETURN
        ENDIF
C CALCULATE NEW {Y}
        DO 9 L=1,LMAX
          DO 10 I=1,NEQ
   10     Y(I,L,N)=Y(I,L,N)+DZ*B(I,L)
C UPDATE FORCE AND MOMENT INTEGRALS
          DFI=(Y(1,L,N)-DZ2*B(1,L))*RB*ABS(DZ)
          FOI(L,N)=FOI(L,N)+DFI
          IF(N.LT.3)EMI(L,N)=EMI(L,N)+ZB*DFI
    9   CONTINUE
    5 CONTINUE
      RETURN
      END
```



```
      SUBROUTINE ESET(Z,N,E,DE)
C SETS DISPLACEMENT/TILT AMPLITUDE, E AND SLOPE DE
C L=1,2,3 CORRESPONDS TO COMPLIMENTARY SOLUTION,TILT,RADIAL DISP RESP. WHEN N<3
C L=1,2   CORRESPONDS TO COMP. SOL. AND AXIAL DISP RESP. WHEN N=3
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION E(3),DE(3),EO(3),DEO(3)
      DATA EO,DEO/0.D0,1.D0,1.D0,0.D0,0.D0,0.D0/
      DO 5 L=1,3
        DE(L)=DEO(L)
    5 E(L)=EO(L)
      IF(N.EQ.3)RETURN
      E(2)=Z
      DE(2)=1.D0
      RETURN
      END
```

```
      SUBROUTINE KBCAL(NOI,IFACE,IDIR,NREG,NRSUB,
     +RE,REC,P1R,OMT,OMDT1,DUT,ZET,
     +IGROT,ALP,SBET,CBET,DELT,ENGP,ZETG,UHG,VHG,
     +RG,ZTG,H,U,V,CK,CB,IER)
C SETS UP BOUNDARY AND CONTINUITY CONDITIONS, SOLVES SECONDARY FLOW PROBLEM
C AND CALCULATES STIFFNESS AND DAMPING COEFFICIENTS.
C V (VI)
C CALLED BY TSEAL
C CALLS DELTP, ESET AND DSOLV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NDZ=201,NDREG=21)
      COMPLEX*16 Y(3,3,3),FOI(3,3),EMI(3,3)
      DIMENSION ALP(NDREG),SBET(NDREG),CBET(NDREG),DELT(NDREG),
     +IGROT(NDREG),UHG(NDZ,NDREG),VHG(NDZ,NDREG)
      DIMENSION NRSUB(NDREG),RG(NDZ,NDREG),ZET(NDREG),ZTG(NDZ,NDREG)
      DIMENSION H(NDZ,NDREG),U(NDZ,NDREG),V(NDZ,NDREG)
      DIMENSION ENGP(NDREG),ZETG(NDREG)
      DIMENSION E(3),DE(3),Y20(3),CK(4,4),CB(4,4)
C INITIAL DISURBANCES IN INLET VELOCITY (V) FOR COMP AND PARTICULAR SOLUTIONS
      DATA Y20/1.D0,0.D0,0.D0/
      RV20=.5D0*REC*P1R
      OMDT=OMDT1
C AVOID INDETERMINACY FOR 0 FREQYENCY DISTURBANCE
      IF(ABS(OMDT).LT.1.D-4)OMDT=1.D-4
C SET UP STARTING CONDITIONS AND LOOPING PARAMETERS FOR SECONDARY FLOW SOLUTION
      KST=1
      KEN=NREG
      IF(IDIR.EQ.-1)THEN
        KST=NREG
        KEN=1
      ENDIF
      NMAX=2+IFACE
      LMAX=3-IFACE
      DO 30 K=KST,KEN,IDIR
        JST=1
        JEN=NRSUB(K)
        IF(IDIR.EQ.-1)THEN
          JST=JEN+1
          JEN=2
        ENDIF
        HJMP=1.D10
        IF(K.NE.KST)HJMP=H(J1,K1)-H(JST,K)
C GET DERIVATIVES FOR FLOW LOSS AT JUMP
        IF(NOI.NE.1)THEN
          CHI=DELTP(RE,V(JST,K),H(JST,K),HJMP,ZET(K))
          CHIH=(DELTP(RE,V(JST,K),H(JST,K)+DUT,HJMP,ZET(K))-CHI)/DUT
          CHIV=(DELTP(RE,V(JST,K)+DUT,H(JST,K),HJMP,ZET(K))-CHI)/DUT
        ENDIF
        DO 5 N=1,NMAX
C SET DISPLACEMENT AMPLITUDES
          CALL ESET(ZTG(JST,K),N,E,DE)
          DO 5 L=1,LMAX
            IF(K.EQ.KST)THEN
C SET UP INITIAL OR CONTINUITY CONDITIONS AT START OF EACH REGION
C (Y) ARE DISTURBANCES IN PRESSURE, AXIAL VELOCITY AND TANGENTIAL VELOCITY
              Y(1,L,N)=(0.D0,0.D0)
              Y(2,L,N)=DCMPLX(Y20(L),0.D0)
              Y(3,L,N)=(0.D0,0.D0)
              FOI(L,N)=(0.D0,0.D0)
              EMI(L,N)=(0.D0,0.D0)
              ELSE
              Y(2,L,N)=V(JST,K)*HJMP/H(JST,K)/H(J1,K1)*E(L)+
     +                 H(J1,K1)/H(JST,K)*Y(2,L,N)
            ENDIF
          IF(NOI.NE.1)Y(1,L,N)=Y(1,L,N)+RV20*(-CHIH*E(L)+CHIV*Y(2,L,N))
    5   CONTINUE
```



```
        K1=K
C STEP THROUGH REGION
        DO 30 J=JST,JEN,IDIR
          J1=J+IDIR
          CALL DSOLV(NOI,IFACE,RE,REC,P1R,OMT,OMDT,DUT,
     +    ALP(K),SBET(K),CBET(K),DELT(K),ENGP(K),ZETG(K),IGROT(K),
     +    RG(J,K),ZTG(J,K),H(J,K),U(J,K),V(J,K),UHG(J,K),VHG(J,K),
     +    RG(J1,K),ZTG(J1,K),H(J1,K),U(J1,K),V(J1,K),
     +    Y,FOI,EMI,IER)
          IF(IER.NE.0)RETURN
   30   CONTINUE
C COMBINE COMP AND PARTICULAR SOL. TO SATISFY P=0 AT DOWNSTREAM BOUNDARY
       DO 40 N=1,NMAX
       DO 40 L=2,LMAX
       FOI(L,N)=-Y(1,L,N)/Y(1,1,N)*FOI(1,N)+FOI(L,N)
   40  EMI(L,N)=-Y(1,L,N)/Y(1,1,N)*EMI(1,N)+EMI(L,N)
       PI=4.D0*ATAN(1.D0)
       PI2=PI/2.D0
       IF(IFACE.EQ.1)THEN
C       EXTRACT STIFFNESS AND DAMPING COEFFICIENTS FOR FACE SEAL
C INITIALIZE STIFFNESS AND DAMPING MATRICES
        DO 41 I=1,4-IFACE
        DO 41 J=1,4-IFACE
          CK(I,J)=0.D0
   41   CB(I,J)=0.D0
C AXIAL FORCE DUE TO AXIAL DISPLACEMENT
        CK(1,1)=2.D0*PI*DREAL(FOI(2,3))
        CB(1,1)=2.D0*PI*DIMAG(FOI(2,3))/OMDT
C MOMENTS DUE TO TILT
        CK(3,3)=PI2*DREAL(EMI(2,1)+EMI(2,2))
        CK(2,2)=CK(3,3)
        CB(3,3)=PI2*DIMAG(EMI(2,1)-EMI(2,2))/OMDT
        CB(2,2)=CB(3,3)
        CK(2,3)=PI2*DIMAG(EMI(2,1)+EMI(2,2))
        CK(3,2)=-CK(2,3)
        CB(2,3)=-PI2*DREAL(EMI(2,1)-EMI(2,2))/OMDT
        CB(3,2)=-CB(2,3)
        ELSE
C       EXTRACT STIFFNESS AND DAMPING COEFFICIENTS FOR CYLINDRICAL SEAL
C MOMENTS DUE TO TILT
        CK(4,4)=PI2*DREAL(EMI(2,1)+EMI(2,2))
        CK(3,3)=CK(4,4)
        CB(4,4)=PI2*DIMAG(EMI(2,1)-EMI(2,2))/OMDT
        CB(3,3)=CB(4,4)
        CK(3,4)=PI2*DIMAG(EMI(2,1)+EMI(2,2))
        CK(4,3)=-CK(3,4)
        CB(3,4)=-PI2*DREAL(EMI(2,1)-EMI(2,2))/OMDT
        CB(4,3)=-CB(3,4)
C MOMENTS DUE TO DISPLACEMENT
        CK(4,1)=PI2*DREAL(EMI(3,1)+EMI(3,2))
        CK(3,2)=-CK(4,1)
        CB(4,1)=PI2*DIMAG(EMI(3,1)-EMI(3,2))/OMDT
        CB(3,2)=-CB(4,1)
        CK(3,1)=PI2*DIMAG(EMI(3,1)+EMI(3,2))
        CK(4,2)=CK(3,1)
        CB(3,1)=-PI2*DREAL(EMI(3,1)-EMI(3,2))/OMDT
        CB(4,2)=CB(3,1)
C FORCES DUE TO TILT
        CK(1,4)=PI2*DREAL(FOI(2,1)+FOI(2,2))
        CK(2,3)=-CK(1,4)
        CB(1,4)=PI2*DIMAG(FOI(2,1)-FOI(2,2))/OMDT
        CB(2,3)=-CB(1,4)
        CK(2,4)=-PI2*DIMAG(FOI(2,1)+FOI(2,2))
        CK(1,3)=CK(2,4)
        CB(2,4)=PI2*DREAL(FOI(2,1)-FOI(2,2))/OMDT
        CB(1,3)=CB(2,4)
```

```
C FORCES DUE TO DISPLACEMENT
        CK(1,1)=PI2*DREAL(FOI(3,1)+FOI(3,2))
        CK(2,2)=CK(1,1)
        CB(1,1)=PI2*DIMAG(FOI(3,1)-FOI(3,2))/OMDT
        CB(2,2)=CB(1,1)
        CK(2,1)=-PI2*DIMAG(FOI(3,1)+FOI(3,2))
        CK(1,2)=-CK(2,1)
        CB(2,1)=PI2*DREAL(FOI(3,1)-FOI(3,2))/OMDT
        CB(1,2)=-CB(2,1)
      ENDIF
      RETURN
      END
```



```
      SUBROUTINE CMATIN(A,B,DETER,N1,M1,ID,N2,INDEX)
C COMPEX MATRIX INVERTER
C CALLED BY COLPC
      IMPLICIT COMPLEX*16(A-H,O-Z)
      DOUBLE PRECISION AMAX
      DIMENSION A(N2,N2),B(N2,1),INDEX(N2,3)
      EQUIVALENCE (IROW,JROW), (ICOLU,JCOLU), (AMAX, T, SWAP)
      M=M1
      N=N1
   10 DETER =(1.D0,0.D0)
      DO 20 J=1,N
   20 INDEX(J,3) = 0
      DO 550 I=1,N
      AMAX=0.0D0
      DO 105 J=1,N
      IF(INDEX(J,3)-1) 60, 105, 60
   60 DO 100 K=1,N
      IF(INDEX(K,3)-1) 80, 100, 715
   80 IF (     AMAX -ABS (A(J,K))) 85, 100, 100
   85 IROW=J
      ICOLU=K
      AMAX =ABS (A(J,K))
  100 CONTINUE
  105 CONTINUE
      IF(AMAX)110,715,110
  110 INDEX(ICOLU,3) = INDEX(ICOLU,3) +1
      INDEX(I,1)=IROW
      INDEX(I,2)=ICOLU
  130 IF (IROW-ICOLU) 140, 310, 140
  140 DETER=-DETER
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLU,L)
  200 A(ICOLU,L)=SWAP
      IF(M) 310, 310, 210
  210 DO 250 L=1, M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLU,L)
  250 B(ICOLU,L)=SWAP
  310  PIVOT   =A(ICOLU,ICOLU)
      IF(PIVOT.EQ.(0.D0,0.D0))GO TO 715
      DETER=DETER*PIVOT
      A(ICOLU,ICOLU)=(1.D0,0.D0)
      DO 350 L=1,N
  350 A(ICOLU,L)=A(ICOLU,L)/PIVOT
      IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLU,L)=B(ICOLU,L)/PIVOT
  380 DO 550 L1=1,N
      IF(L1-ICOLU) 400, 550, 400
  400 T=A(L1,ICOLU)
      A(L1,ICOLU)=(0.D0,0.D0)
      IF(T.EQ.(0.D0,0.D0))GO TO 550
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLU,L)*T
      IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLU,L)*T
  550 CONTINUE
  600 DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLU=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
```

```
      A(K,JROW)=A(K,JCOLU)
      A(K,JCOLU)=SWAP
  705 CONTINUE
  710 CONTINUE
      ID =1
  740 RETURN
  715 ID =2
      DETER=(0.D0,0.D0)
      RETURN
      END
```



```
      SUBROUTINE MATINV(A,B,DETER,N1,M1,ID,N2,INDEX)
C REAL MATRIX INVERSION ROUTINE
C CALLED BY HOME COLP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION A(N2,N2),B(N2,1),INDEX(N2,3)
      EQUIVALENCE (IROW,JROW), (ICOLU,JCOLU), (AMAX, T, SWAP)
      M=M1
      N=N1
   10 DETER = 1.D0
      DO 20 J=1,N
   20 INDEX(J,3) = 0
      DO 550 I=1,N
      AMAX=0.D0
      DO 105 J=1,N
      IF(INDEX(J,3)-1) 60, 105, 60
   60 DO 100 K=1,N
      IF(INDEX(K,3)-1) 80, 100, 715
   80 IF (     AMAX -ABS (A(J,K))) 85, 100, 100
   85 IROW=J
      ICOLU=K
      AMAX = ABS (A(J,K))
  100 CONTINUE
  105 CONTINUE
      IF(AMAX)110,715,110
  110 INDEX(ICOLU,3) = INDEX(ICOLU,3) +1
      INDEX(I,1)=IROW
      INDEX(I,2)=ICOLU
  130 IF (IROW-ICOLU) 140, 310, 140
  140 DETER=-DETER
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLU,L)
  200 A(ICOLU,L)=SWAP
      IF(M) 310, 310, 210
  210 DO 250 L=1, M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLU,L)
  250 B(ICOLU,L)=SWAP
  310 PIVOT   =A(ICOLU,ICOLU)
      IF(PIVOT.EQ.0.D0)GO TO 715
      DETER=DETER*PIVOT
      A(ICOLU,ICOLU)=1.D0
      DO 350 L=1,N
  350 A(ICOLU,L)=A(ICOLU,L)/PIVOT
      IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLU,L)=B(ICOLU,L)/PIVOT
  380 DO 550 L1=1,N
      IF(L1-ICOLU) 400, 550, 400
  400 T=A(L1,ICOLU)
      A(L1,ICOLU)=0.D0
      IF(T)430,550,430
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLU,L)*T
      IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLU,L)*T
  550 CONTINUE
  600 DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
      JCOLU=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLU)
```

```
      A(K,JCOLU)=SWAP
  705 CONTINUE
  710 CONTINUE
      ID =1
  740 RETURN
  715 ID =2
      DETER=0.D0
      RETURN
      END
```

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY (*Leave blank*) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | October 2005 | Final Contractor Report |

**4. TITLE AND SUBTITLE**

Source Listings for Computer Code SPIRALI

Incompressible, Turbulent Spiral Grooved Cylindrical and Face Seals

**5. FUNDING NUMBERS**

WU–506–42–31–00
WU–590–21–11–00
NAS3–25644

**6. AUTHOR(S)**

Jed A. Walowit and Wilbur Shapiro

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Mechanical Technology, Inc.
968 Albany Shaker Road
Latham, New York 12110

**8. PERFORMING ORGANIZATION REPORT NUMBER**

E–13616

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

National Aeronautics and Space Administration
Washington, DC 20546–0001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

NASA CR—2003-212360
MTI 95TM6

**11. SUPPLEMENTARY NOTES**

Jed A. Walowit, Jed A. Walowit, Inc., 4 Cypress Point, Clifton Park, New York 12065; and Wilbur Shapiro, Mechanical Technology, Inc., Latham, New York. Project Manager, Anita D. Liang, Aeropropulsion Projects Office, NASA Glenn Research Center, organization code P, 216–977–7439.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Restriction changed to Unclassified/Unlimited on July 27, 2005, by authority of the NASA Glenn Research Center, Structures Division.

**Export Administration Regulations (EAR) Notice**

This document contains information within the purview of the Export Administration Regulations (EAR), 15 CFR 730–774, and is export controlled. It may not be transferred to foreign nationals in the U.S. or abroad without specific approval of a knowledgeable NASA export control official, and/or unless an export license/license exception is obtained/available from the Bureau of Industry and Security, United States Department of Commerce. Violations of these regulations are punishable by fine, imprisonment, or both.

Unclassified-Unlimited
Subject Category: 34
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** ***(Maximum 200 words)***

This is the source listing of the computer code SPIRALI which predicts the performance characteristics of incompressible cylindrical and face seals with or without the inclusion of spiral grooves. Performance characteristics include load capacity (for face seals), leakage flow, power requirements and dynamic characteristics in the form of stiffness, damping and apparent mass coefficients in 4 degrees of freedom for cylindrical seals and 3 degrees of freedom for face seals. These performance characteristics are computed as functions of seal and groove geometry, load or film thickness, running and disturbance speeds, fluid viscosity, and boundary pressures.

**14. SUBJECT TERMS**

Face seals; Seals; Cylindrical seals; Leakage; Incompressible; Spiral-grooved; Turbulent; Source listing

**15. NUMBER OF PAGES**

28

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |